CSCI1680
Network Layer: IP & Forwarding

Nick DeMarinis

Administivia

- Snowcast: due Wednesday (9/27)

- IP project: out Thursday, fill out group preference form by Thursday 11:59pm

- HW1: out later today, due next Thurs
  - Some practice for IP!

Today

Start of network layer

- Network layer: Internet Protocol (IP) (v4)
- Mechanics of IP forwarding
- Intro to IP project

Last time: link-layer

- Hosts connect to network via <u>interfaces</u>
- Every interface has a link-layer address
  Ethernet/Wifi: MAC address (`0c:45:22:c1:be:03`)
- Switches, Wifi APs: in-network devices that forward packets between nodes

<u>Mental model for the link layer</u>
=> How to connect hosts on a "small" network
=> Given link-layer address, know how to reach host on your network

Layers, Services, Protocols

|	
Application	Service: user-facing application. Application-defined messages
Transport	Service: multiplexing applications Reliable byte stream to other node (TCP), Unreliable datagram (UDP)
Network	Service: move packets to any other node in the network Internet Protocol (IP)
Link	Service: move frames to other node across link. May add reliability, medium access control
Physical	Service: move bits to other node across link

Internet Protocol (IP) Goals

How to connect *everyone*?

- Glue lower-level networks together

- A network of networks!

- Router: device that forwards packets between *networks*

=> Doesn't that sound like switching?
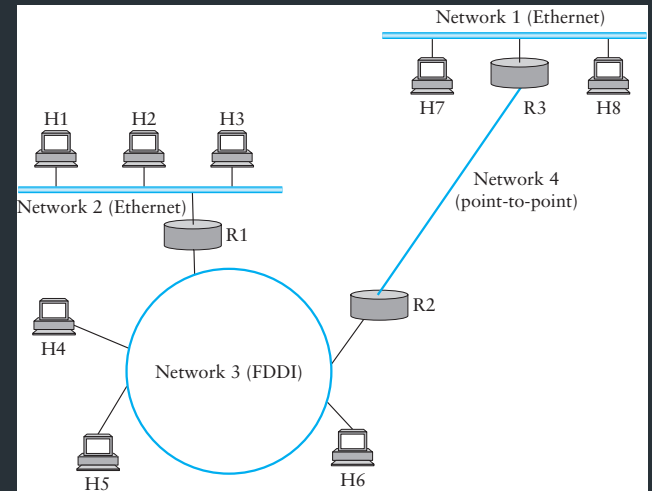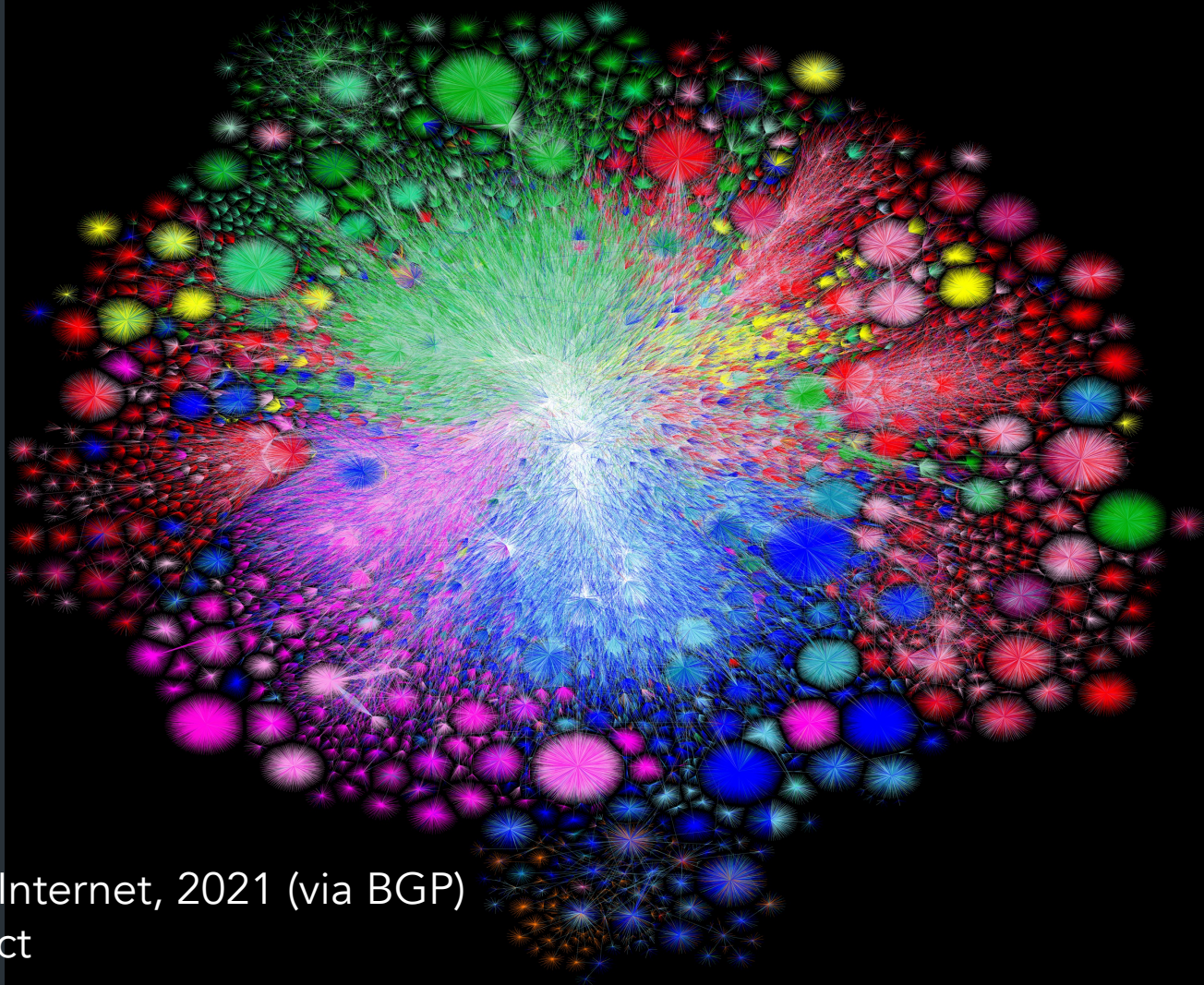
Color Chart
- North America (ARIN)
- Europe (RIPE)
- Asia Pacific (APNIC)
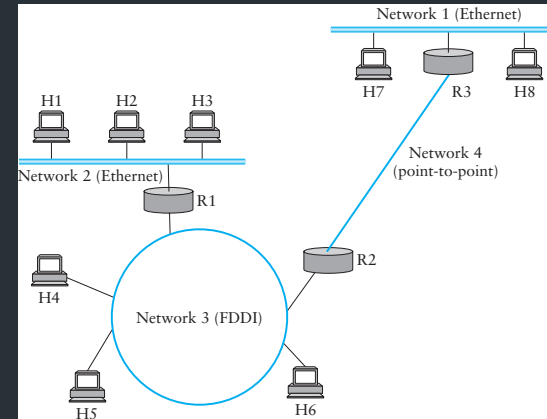- Latin America (LANIC)
- Africa (AFRINIC)
- Backbone
- US Military

Map of the Internet, 2021 (via BGP)
OPTE project

New Challenges

- Networks are heterogeneous (eg. Wifi vs. Ethernet)
  - Different frame formats
  - Some are more reliable than others
  - Different packet sizes/bandwidths
- Scaling: link-layer strategies don't work!

What came before the Internet?

The (landline) telephone network

(Plain Old Telephone Service (POTS))

At the time the internet was forming (~1960s), landline phones were a big network that connected lots of end-users (160M worldwide)

Why should we care?
⇒ Provides a useful comparison
⇒ Understand some of the design goals of the time

Early telephone networks

Used circuit switching: set up whole path for call beforehand

• Does it scale?

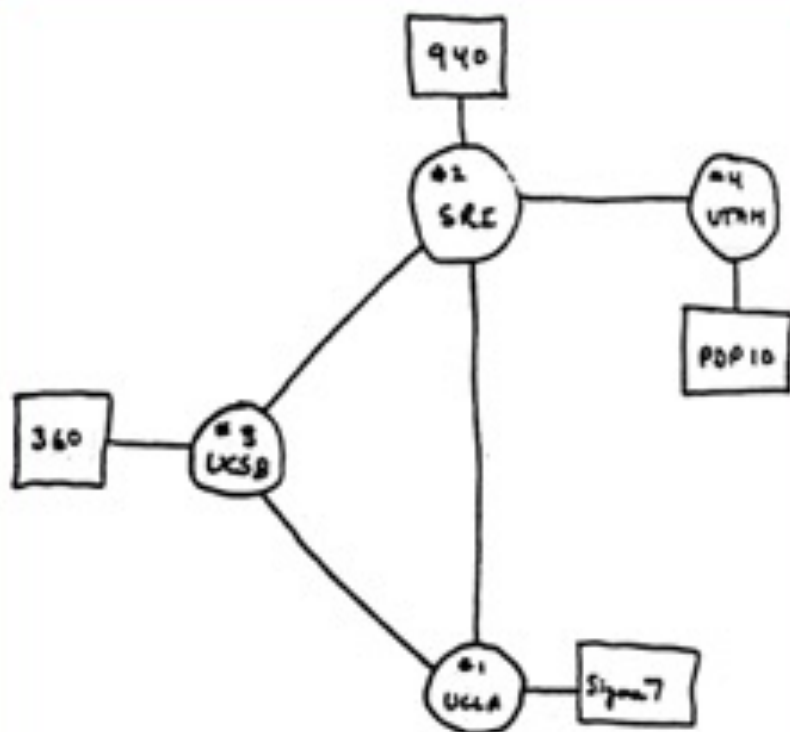# A Bit of History

Early Packet switched networks: Arpanet's IMPs
- Late 1960's => RFC 1, 1969!
- Reliable network with many features we know today

- Build apps with Network Control Program (NCP)
  - Built on reliable IMPs
  - Used by programs like telnet, mail, file transfer

What about when network isn't reliable?

THE ARPA NETWORK
DEC 1969

Abb. 4 ARPA NETwork, topologische Karte. Stand Juni 1974.

How to make such a protocol?

- How to deal with heterogeneous networks?
- How to find hosts?
- Should messages be reliable or unreliable?
- What to do when a device joins/leaves?
- …

Big concerns
⇒ Not every application needs all features
⇒ Can't assume much functionality from (heterogeneous link layer)

1974: TCP/IP Introduced

- Vint Cerf, Robert Kahn build protocol to replace NCP
- Initial design: single protocol providing a reliable pipe

- Eventually, separated into different protocols we know today
  - IP: basic datagram service among hosts
  - TCP: reliable transport
  - UDP: unreliable *multiplexed* datagram service

SAN FRANCISCO BAY AREA PACKET RADIO NET

ARPANET

BOLT BERANEK AND NEWMAN CAMBRIDGE, MASSACHUSETTS

NORWEGIAN DEFENSE RESEARCH ESTABLISHMENT KJELLER, NORWAY

NORSAR TIP

PACKET RADIO NET

SRI

ETAM, WEST VIRGINIA EARTH STATION

INTELSAT IV-A

SRI MENLO PARK, CALIFORNIA

SRI VAN

GOONHILLY DOWNS, ENGLAND EARTH STATION

TANUM, SWEDEN EARTH STATION

ATLANTIC PACKET SATELLITE NET

UNIVERSITY COLLEGE LONDON LONDON, ENGLAND

LONDON TIP

LEGEND

R PACKET RADIO REPEATER

I ARPANET IMP

T ARPANET TIP

S SATELLITE IMP

G INTERNETWORK GATEWAY

S/G PACKET RADIO STATION INTERNETWORK GATEWAY

ISI-C

UNIVERSITY OF SOUTHERN CALIFORNIA INFORMATION SCIENCES INSTITUTE MARINA DEL REY, CALIFORNIA

.... PATH OF PACKETS

18

IP's Decisions

- Connectionless, packet-switched network

- "Best-effort" service: other layers add reliability if you need it

How to reach hosts?
- Common message format: IP header
- Every host identified by an IP address

IP's Decisions

- Connectionless, packet-switched network

  => Routers are "simple" => no connection state

- "Best-effort" service: other layers add reliability if you need it

  => Packets might be dropped, reordered, delayed, …

How to reach hosts?

- Common message format: IP header
- Every host identified by an IP address

An excellent read

David D. Clark, "The design Philosophy of the DARPA Internet Protocols", 1988

- Primary goal: multiplexed utilization of existing interconnected networks
- Other goals:
  - Communication continues despite loss of networks or gateways
  - Support a variety of communication services
  - Accommodate a variety of networks
  - Permit distributed management of its resources
  - Be cost effective
  - Low effort for host attachment
  - Resources must be accountable

The Internet Protocol
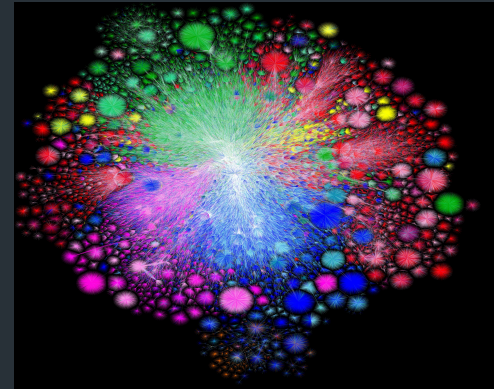
IP runs on all hosts and routers

- _Addressing_: how we name nodes in an IP network
- Provides _forwarding_: how routers move packets based on the destination address
- _(later)_ _Routing_: how routers build forwarding rules
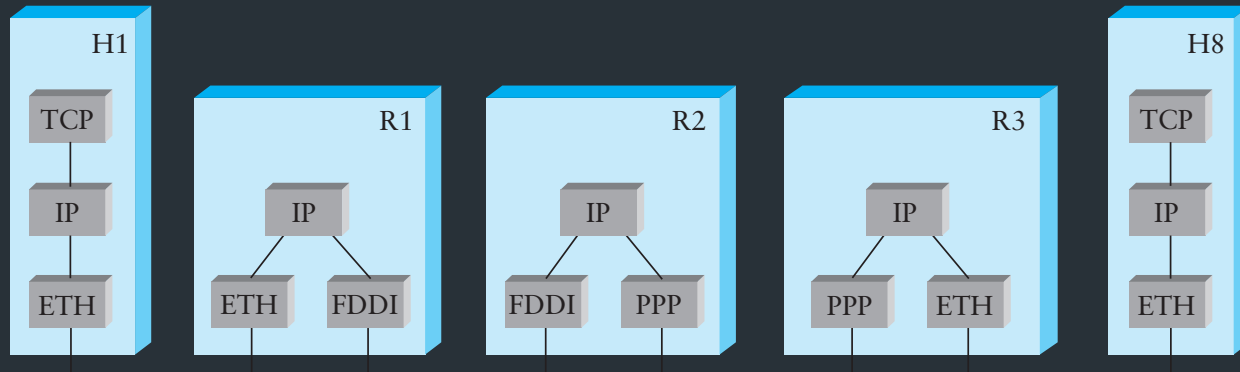
IP Addressing

Overview

- Unique number to identify "all" hosts on the Internet

- A number with structure => the number tells the network where the host is

Analogy: back to phones

Telephone numbers have a structure to them

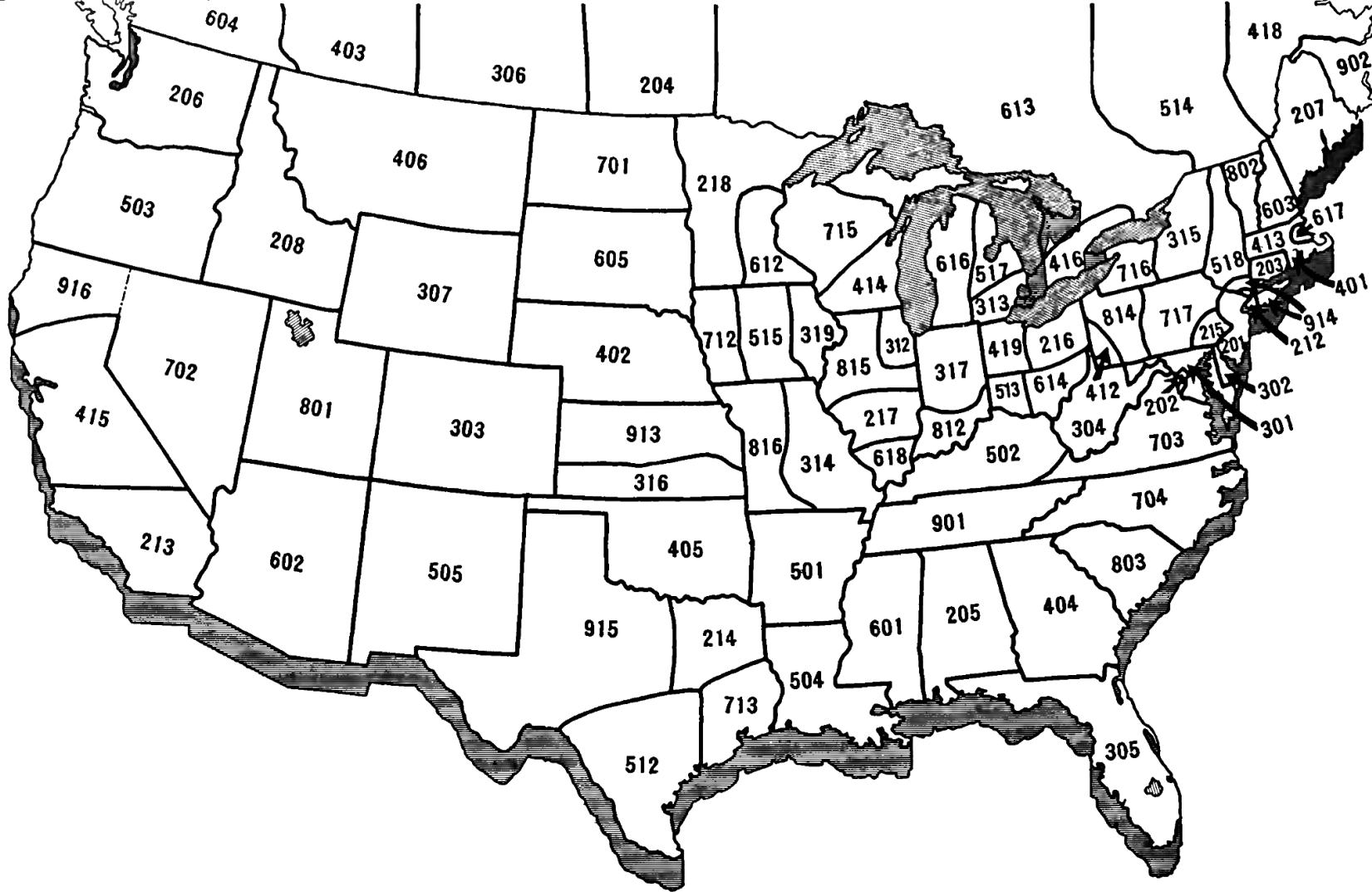+ 1 401 863 1000

Analogy: back to phones

Telephone numbers have a structure to them

+ 1 401 863 1000

+1 212 555 4253

Part of the number tells where you are!
(or at least it did before cell phones)
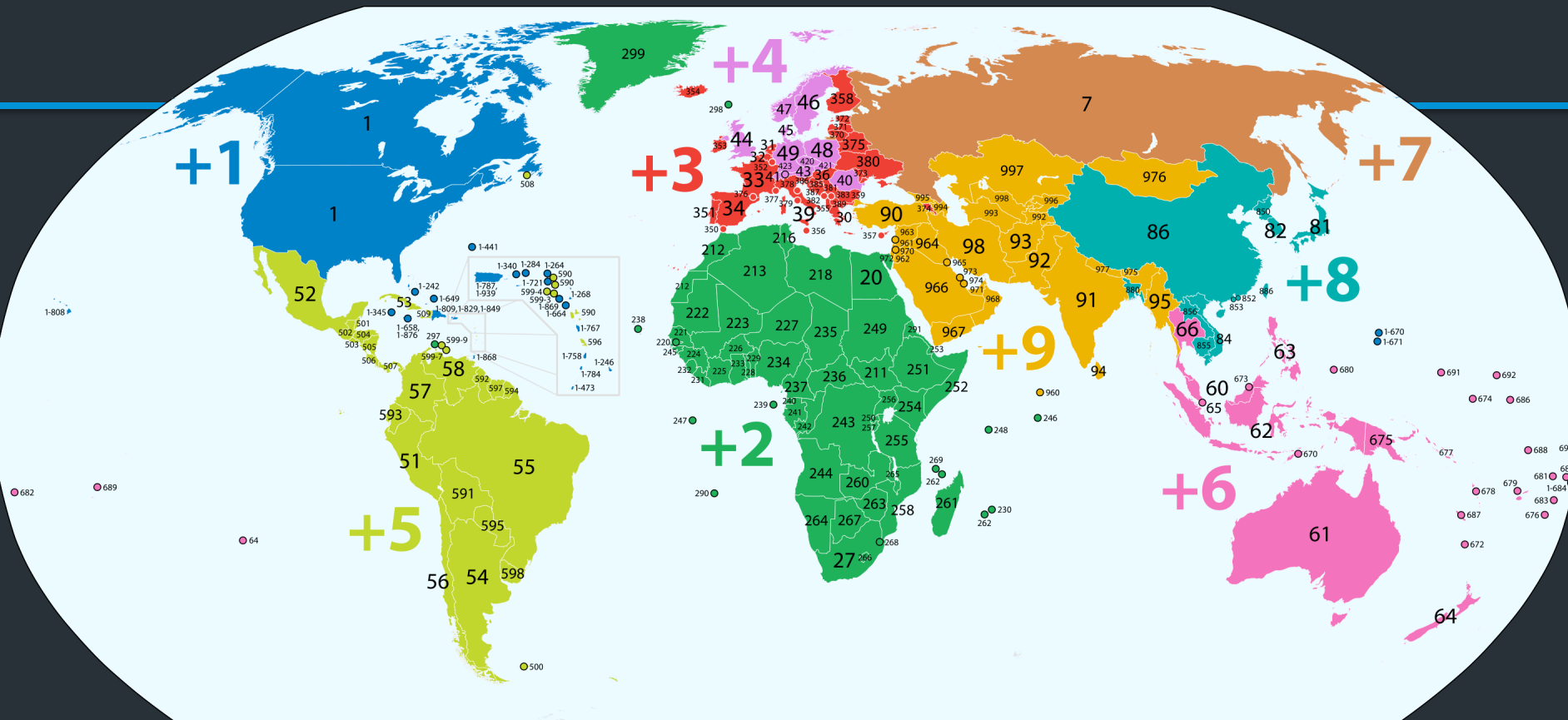
… and not all numbers are the same length!

IP Addressing

IP Version 4: Each address is a 32-bit number:

128.148.16.7

128.148.16.7

```
10000000 10010100 00010000 00000111
```

32 bits => 2^{32} possible addresses… problem?

<u>Notation</u>
• Write each byte ("octet") as a decimal number
• This is called "dotted decimal" or "dotted quad" notation

IP Addressing

An IP address identifies…

- *Who* a host is: A unique number

- *Where* it is on the Internet

- Networks are allocated ranges of IPs by global authority (ICANN)
  - Further subdivided by regions, ISPs, …
- Some IPs have special uses (eg. 127.0.0.1)

128.148.16.7
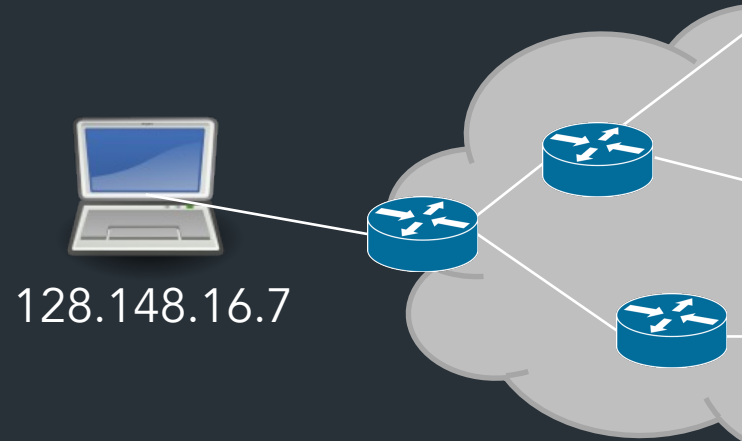
eg. Brown owns 128.148.xxx.xxx, 138.16.xxx.xxx

*ICANN (Internet Corporation for Assigned Names and Numbers)

IP Addressing

Brown owns the range:

128.148.xxx.xxx

10000000 10010100 xxxxxxxx xxxxxxxx

Network part
Identifies Brown (to the Internet)

Host part
Denotes individual hosts
within the Brown Network

A typical configuration
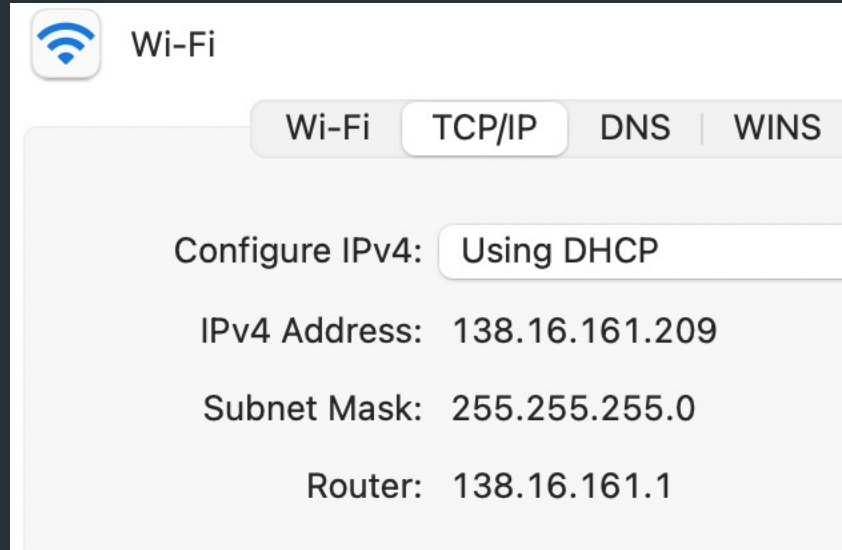
IP Addressing

A network can designate IP addresses for its own hosts within its address range

For 128.148.xxx.xxx:

`10000000 10010100 xxxxxxxx xxxxxxxx`

Brown uses the the prefix 128.148.0.0/16

Some other ways to write this:
128.148/16
128.148.0.0 + subnet mask 255.255.0.0

Brown

128.148.5.1

128.148.10.100

128.148.200.5
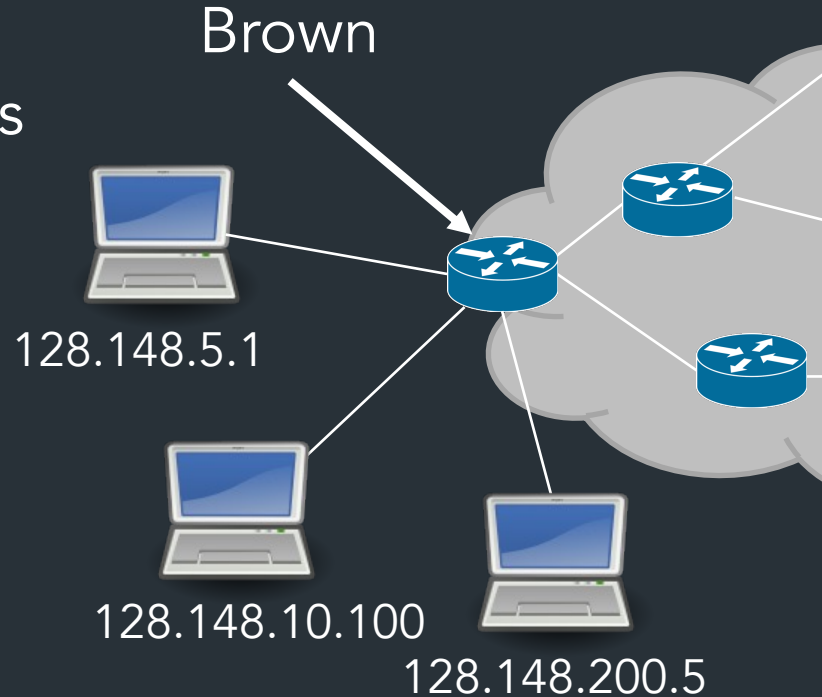
Common prefixes

1.2.0.0/16   **00000001 00000010 xxxxxxxx xxxxxxxx**

8.0.0.0/8   **00001000 xxxxxxxx xxxxxxxx xxxxxxxx**

123.10.1.0/24   **01111011 00001010 00000001 xxxxxxxx**

201.112.10.200/30   **11001001 01110000 00001010 110010xx**

Example

How many addresses are in the network 1.1.0.0/20?

*How do we move
packets between networks?*

How IP forwarding works

Assume:

- Communicating on same network is easy—this is the link-layer's job!
- Can map IP addresses to MAC addresses (more on this later)

How to reach an address *outside* this network?

> Send packets to a <u>router</u>, which <u>forwards</u> IP packets to other networks
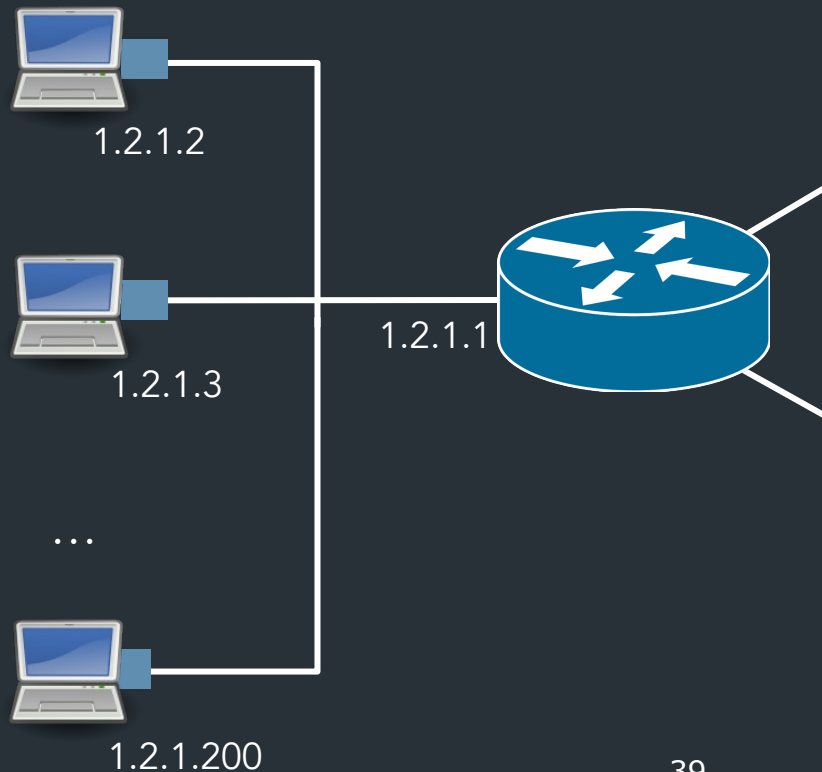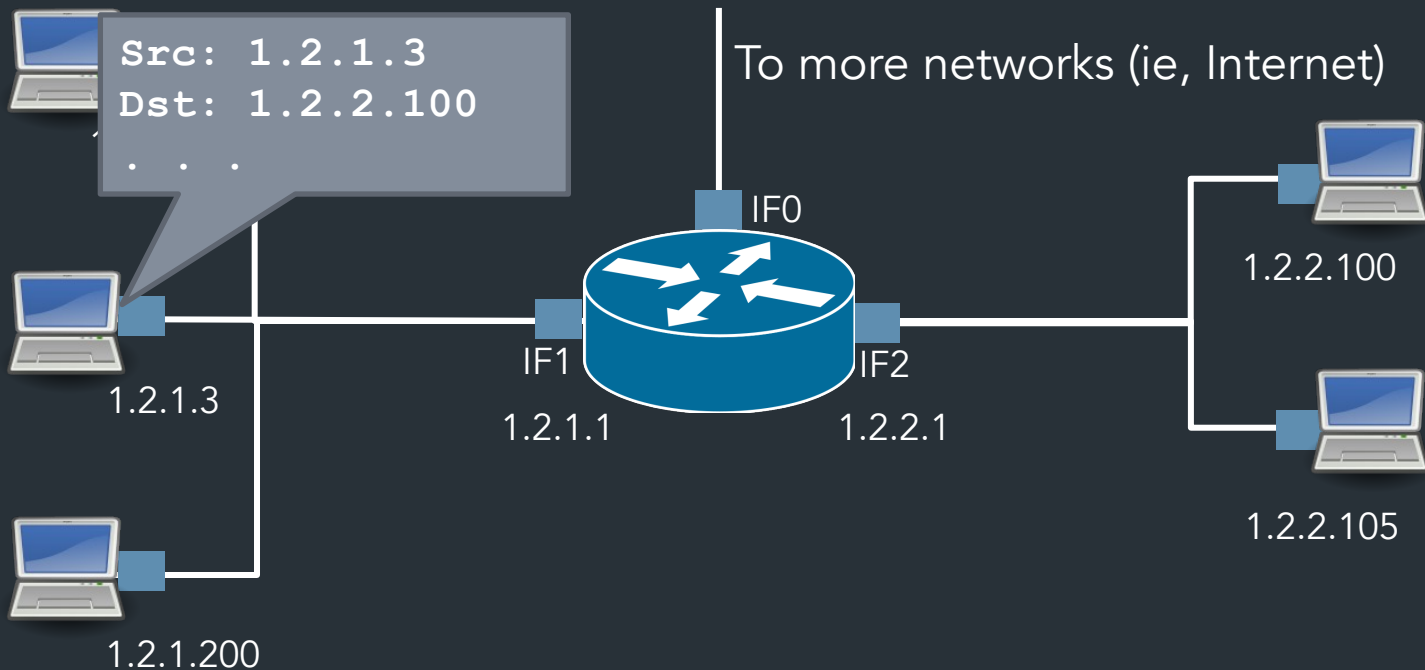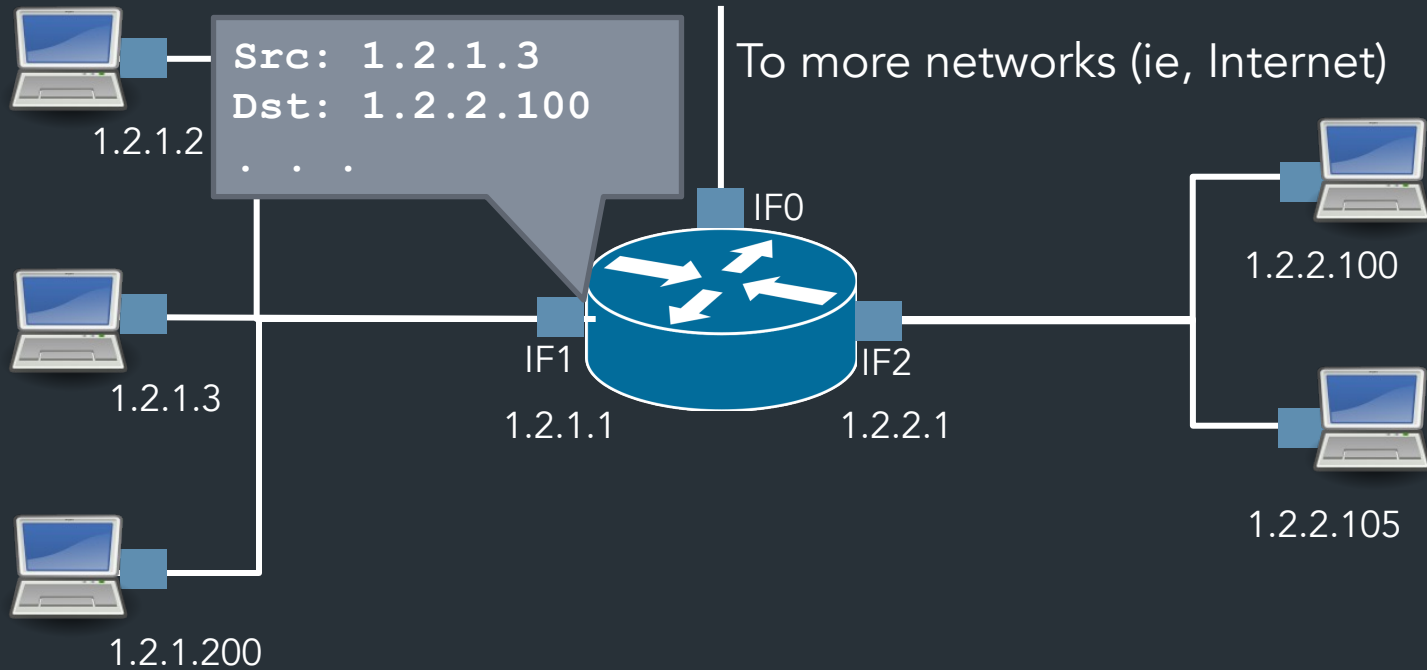
1.2.1.2

1.2.1.3

…

1.2.1.200

1.2.1.1

Forwarding IP packets

Src: 1.2.1.3
Dst: 1.2.2.100
. . .

To more networks (ie, Internet)

IF0

IF1

IF2

1.2.1.3

1.2.1.1

1.2.2.1

1.2.1.200

1.2.2.100

1.2.2.105

Forwarding IP packets

Src: 1.2.1.3
Dst: 1.2.2.100
. . .

1.2.1.2

1.2.1.3

1.2.1.200

To more networks (ie, Internet)

IF0

IF1
1.2.1.1

IF2
1.2.2.1

1.2.2.100

1.2.2.105

Forwarding IP packets

Src: 1.2.1.3
Dst: 1.2.2.100
. . .

1.2.1.2

1.2.1.3

1.2.1.200

To more networks (ie, Internet)

IF0

IF1
1.2.1.1

IF2
1.2.2.1

1.2.2.100

1.2.2.105

Prefix	Interface

Forwarding IP packets

Src: 1.2.1.3
Dst: 1.2.2.100
. . .

To more networks (ie, Internet)

1.2.1.2

1.2.1.3

1.2.1.200

IF0

IF1
1.2.1.1

IF2
1.2.2.1

1.2.2.100

1.2.2.105

Prefix	Interface
1.2.1.0/24	IF1
1.2.2.0/24	IF2
<everything else>	(IF0)

43
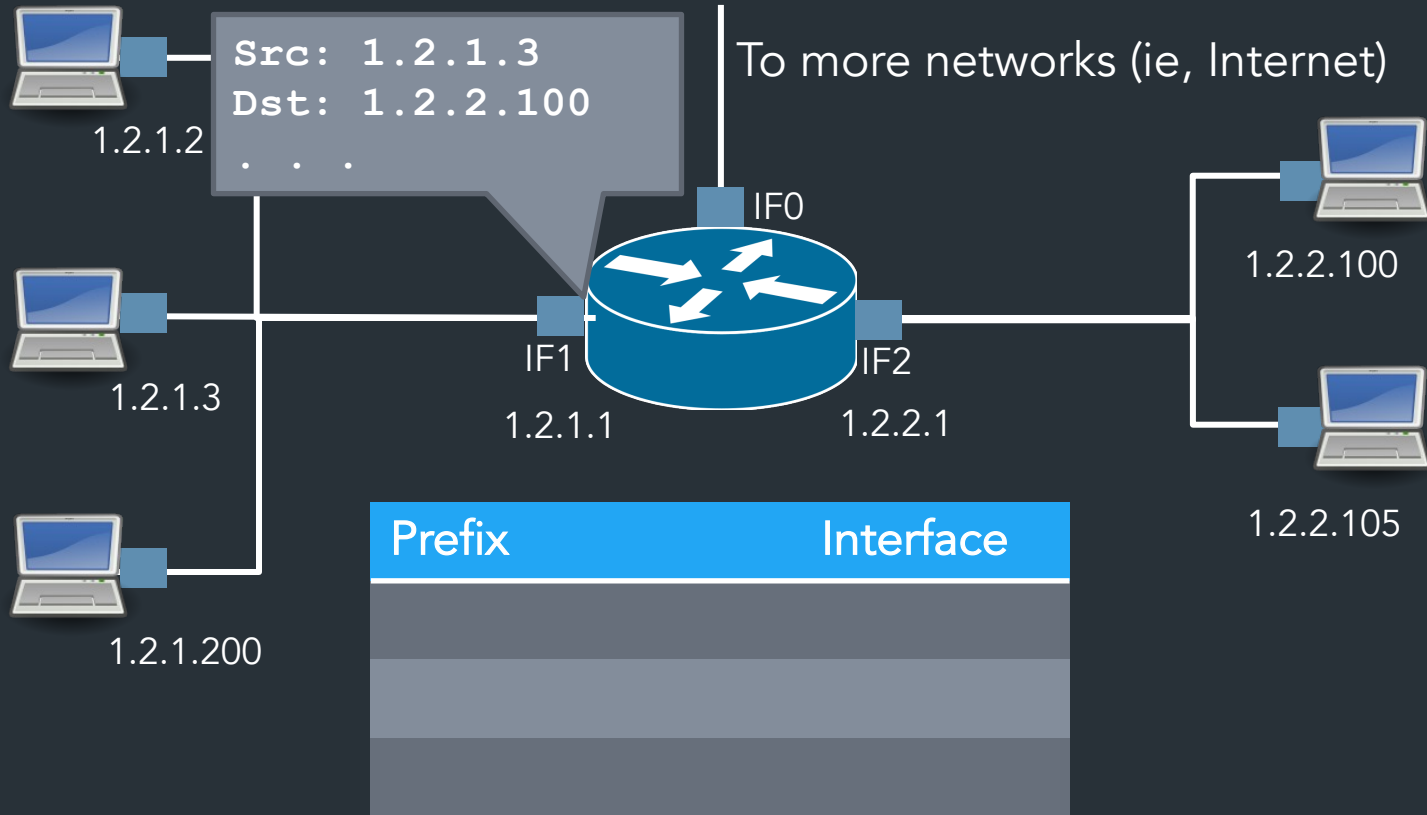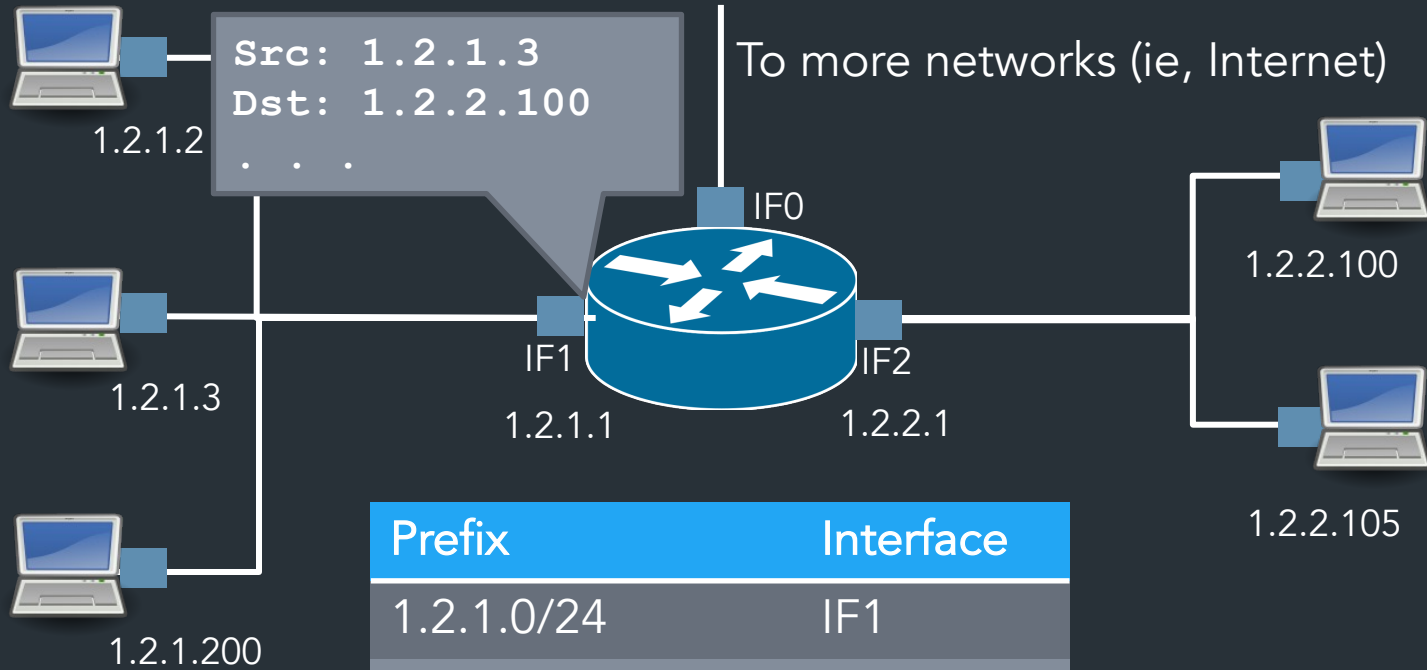
Wait, what happens at the link layer?

IF0

IF1

IF2

1.2.1.3

1.2.1.1

1.2.2.1

1.2.2.100
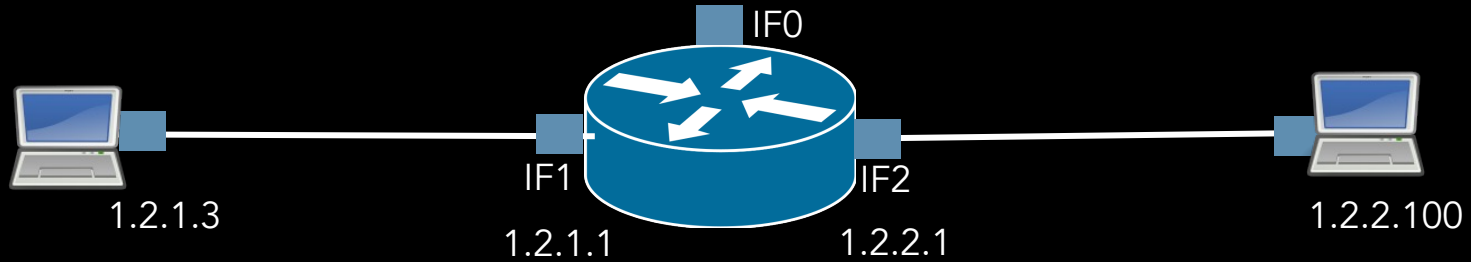
What about the rest?

How to reach networks that aren't directly connected?

To more networks
(ie, Internet)

Prefix	Interface
1.2.1.0/24	IF1
1.2.2.0/24	IF2
<everything else>	IF0

IF0 8.0.0.1

1.2.1.0/24   IF1   IF2   1.2.2.0/24
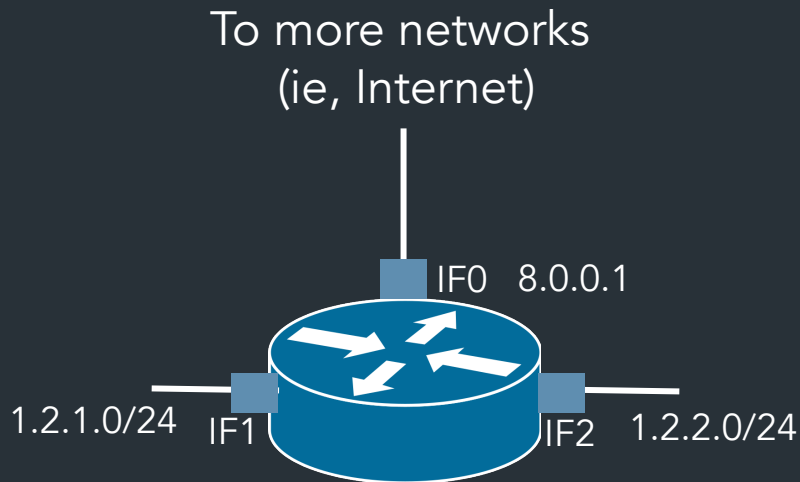
What about the rest?

- Need "next hop" IP: another router that knows about other networks
  - How to reach it? Check table again!
- "Default gateway": where to send to reach anything not in the table

Prefix	IF/Next hop
1.2.1.0/24	IF1
1.2.2.0/24	IF2
8.0.0.0/30	IF0
128.148.0.0/16	1.2.1.5
Default	8.0.0.2

8.0.0.2

8.0.0.1   IF0

1.2.1.0/24   IF1   IF2   1.2.2.0/24
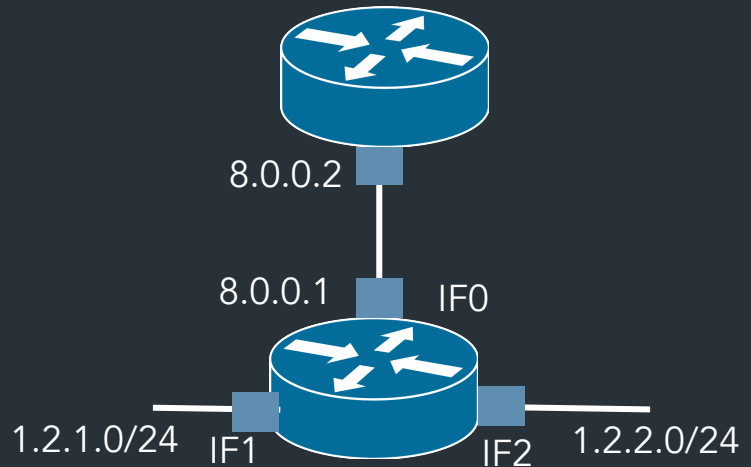
The forwarding table

Exploits hierarchical structure of addresses: know how to reach _networks_, not individual hosts

Prefix	IF/Next hop
1.2.1.0/24	IF1
1.2.2.0/24	IF2
8.0.0.0/30	IF0
128.148.0.0/16	1.2.1.5
Default	8.0.0.2

- Table is keyed is a network prefix, not a whole address
- Select best prefix with _longest prefix matching_ (more on this later)

A forwarding table

```
# ip route
127.0.0.0/8 via 127.0.0.1 dev lo
172.17.44.0/24 dev enp7s0  proto kernel  scope link  src 172.17.44.22  metric 204
default via 172.17.44.1 dev eth0 src 172.17.44.22  metric 204
```

*How do we move
packets between networks?*

IP forwarding

Given a packet, decide where to send it

Prefix	Interface/Next hop

A forwarding table (my laptop)

```
deemer@ceres ~ % ip route default via 10.3.128.1 dev wlp2s0
10.3.128.0/18 dev wlp2s0 proto dhcp scope link src 10.3.135.44 metric 3003
172.18.0.0/16 dev docker0 proto kernel scope link src 172.18.0.1
192.168.1.0/24 dev enp0s31f6 proto kernel scope link src 192.168.1.1
```

Forwarding: examples

Prefix	Interface/Next hop

Routing based on <u>networks</u>

A routing table

```
R6#sh ip ro
Gateway of last resort is 108.34.215.1 to network 0.0.0.0

S*     0.0.0.0/0 [1/0] via 108.34.215.1
       10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
C         10.1.0.0/24 is directly connected, wlan-ap0
L         10.1.0.2/32 is directly connected, wlan-ap0
O IA      10.1.44.1/32 [110/1001] via 10.20.30.33, 3w4d, Tunnel0
C         10.1.48.0/24 is directly connected, Loopback0
L         10.1.48.1/32 is directly connected, Loopback0
C         10.20.30.32/31 is directly connected, Tunnel0
L         10.20.30.32/32 is directly connected, Tunnel0
       108.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C         108.34.215.0/24 is directly connected, GigabitEthernet0/0
L         108.34.215.208/32 is directly connected, GigabitEthernet0/0
       172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C         172.16.98.0/24 is directly connected, Vlan98
L         172.16.98.1/32 is directly connected, Vlan98
       172.17.0.0/16 is variably subnetted, 6 subnets, 3 masks
O IA      172.17.44.0/24 [110/1001] via 10.20.30.33, 3w4d, Tunnel0
C         172.17.48.0/24 is directly connected, Vlan20
L         172.17.48.1/32 is directly connected, Vlan20
C         172.17.49.0/25 is directly connected, Vlan50
```

A routing table

```
R6#sh ip ro
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override


Gateway of last resort is 108.34.215.1 to network 0.0.0.0


S*     0.0.0.0/0 [1/0] via 108.34.215.1
       10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
C         10.1.0.0/24 is directly connected, wlan-ap0
L         10.1.0.2/32 is directly connected, wlan-ap0
O IA      10.1.44.1/32 [110/1001] via 10.20.30.33, 3w4d, Tunnel0
C         10.1.48.0/24 is directly connected, Loopback0
L         10.1.48.1/32 is directly connected, Loopback0
C         10.20.30.32/31 is directly connected, Tunnel0
L         10.20.30.32/32 is directly connected, Tunnel0
```
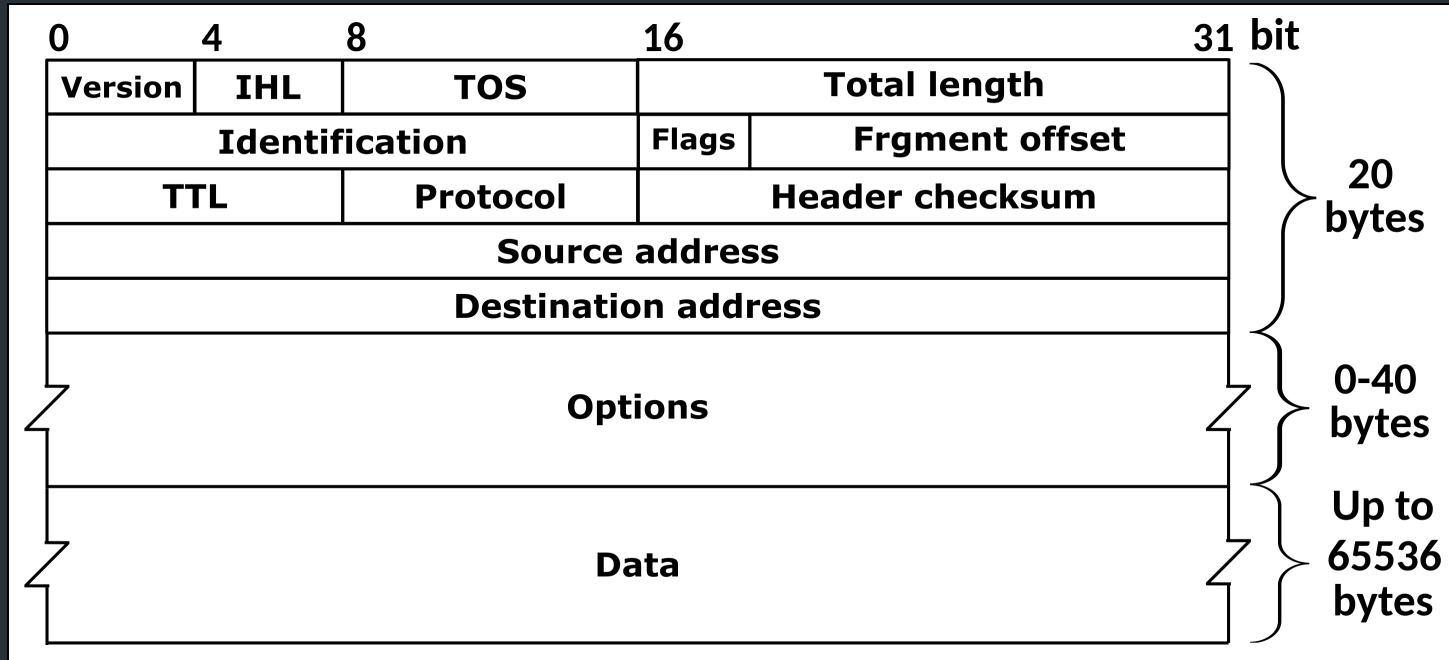
A large table

```
rviews@route-server.ip.att.net>show route table inet.0 active-path

inet.0: 866991 destinations, 13870153 routes (866991 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both


0.0.0.0/0              *[Static/5] 5w0d 19:43:09
                        > to 12.0.1.1 via em0.0
1.0.0.0/24             *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238
                          AS path: 7018 3356 13335 I, validation-state: valid
                        > to 12.0.1.1 via em0.0
1.0.4.0/22             *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238
                          AS path: 7018 3356 4826 38803 I, validation-state: valid
                        > to 12.0.1.1 via em0.0
1.0.4.0/24             *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238
                          AS path: 7018 3356 4826 38803 I, validation-state: valid
                        > to 12.0.1.1 via em0.0
1.0.5.0/24             *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238
                          AS path: 7018 3356 4826 38803 I, validation-state: valid
                        > to 12.0.1.1 via em0.0
1.0.6.0/24             *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238
                          AS path: 7018 3356 4826 38803 I, validation-state: valid
                        > to 12.0.1.1 via em0.0
```

How does forwarding actually work?

The IPv4 Header

0	4	8	16	31 bit	
Version	IHL	TOS	Total length		20 bytes
Identification			Flags	Frgment offset	
TTL		Protocol	Header checksum		
Source address					
Destination address					
Options					0-40 bytes
Data					Up to 65536 bytes

Most Important fields

- <u>Version</u>: 4 for IPv4 packets, 6 for IPv6
- <u>Source address</u>: where the packet came from
- <u>Destination address</u>: where the packet is going

(continued…)

More important fields

- <u>TTL (time-to-live)</u>: decremented each hop
  - Can prevent forwarding loops (and do other stuff…)
- <u>Checksum</u>: computed over <u>header</u> (very weak!)
- <u>Protocol identifier</u>: describes what's in the packet
  - 6: TCP, 17: UDP, 1: ICMP, …
  - Defines the type of the payload

Less important fields

- Header length: in 32-bit units
  - \>5 implies use of IP options
  - Almost all routers ignore IP options
- Fragmentation
  - Network can fragment a packet if next link requires a small frame
  - Most routers don't fragment (or reassemble fragments)
- We won't talk about…
  - Type of Service (TOS): basic traffic classification
  - Identifier: might have special meaning on some networks

Forwarding steps

What does a router do when it receives a packet?

Forwarding mechanics

When an IP packet arrives at a host/router:

- Is it valid? Verify checksum over *header*

- **Is it for me?** If dest IP == your address, send to OS

- If not, where should it go?
    - Consult forwarding table => find next hop
    - Decrement TTL
    - Send packet to next hop

Traceroute

- When TTL reaches 0, router may send back an error
  - ICMP TTL exceeded
- If it does, we can identify a path used by a packet!

Traceroute example

```
[deemer@Warsprite ~]$ traceroute -q 1 google.com
traceroute to google.com (142.251.40.174), 30 hops max, 60 byte packets
 1  router1-nac.linode.com (207.99.1.13)  0.621 ms
 2  if-0-1-0-0-0.gw1.cjj1.us.linode.com (173.255.239.26)  0.499 ms
 3  72.14.222.136 (72.14.222.136)  0.949 ms
 4  72.14.222.136 (72.14.222.136)  0.919 ms
 5  108.170.248.65 (108.170.248.65)  1.842 ms
 6  lga25s81-in-f14.1e100.net (142.251.40.174)  1.812 ms
```

Traceroute example

```
[deemer@Warsprite ~]$ traceroute -q 1 amazon.co.uk
traceroute to amazon.co.uk (178.236.7.220), 30 hops max, 60 byte packets
 1   router2-nac.linode.com (207.99.1.14)  0.577 ms
 2   if-11-1-0-1-0.gw2.cjj1.us.linode.com (173.255.239.16)  0.461 ms
 3   ix-et-2-0-2-0.tcore3.njy-newark.as6453.net (66.198.70.104)  1.025 ms
 4   be3294.ccr41.jfk02.atlas.cogentco.com (154.54.47.217)  2.938 ms
 5   be2317.ccr41.lon13.atlas.cogentco.com (154.54.30.186)  69.725 ms
 6   be2350.rcr21.b023101-0.lon13.atlas.cogentco.com (130.117.51.138)  69.947 ms
 7   a100-row.demarc.cogentco.com (149.11.173.122)  71.639 ms
 8   150.222.15.28 (150.222.15.28)  78.217 ms
 9   150.222.15.21 (150.222.15.21)  84.383 ms
10   *
11   150.222.15.4 (150.222.15.4)  74.529 ms
                                        . . .
30   178.236.14.162 (178.236.14.162)  83.659 ms
```

Demo: IP project

The IPv4 Header

Version	IHL	TOS	Total length		
Identification			Flags	Frgment offset	
TTL		Protocol	Header checksum		
Source address					
Destination address					
Options					
Data					

0   4   8   16   31 bit

20 bytes

0-40 bytes

Up to 65536 bytes
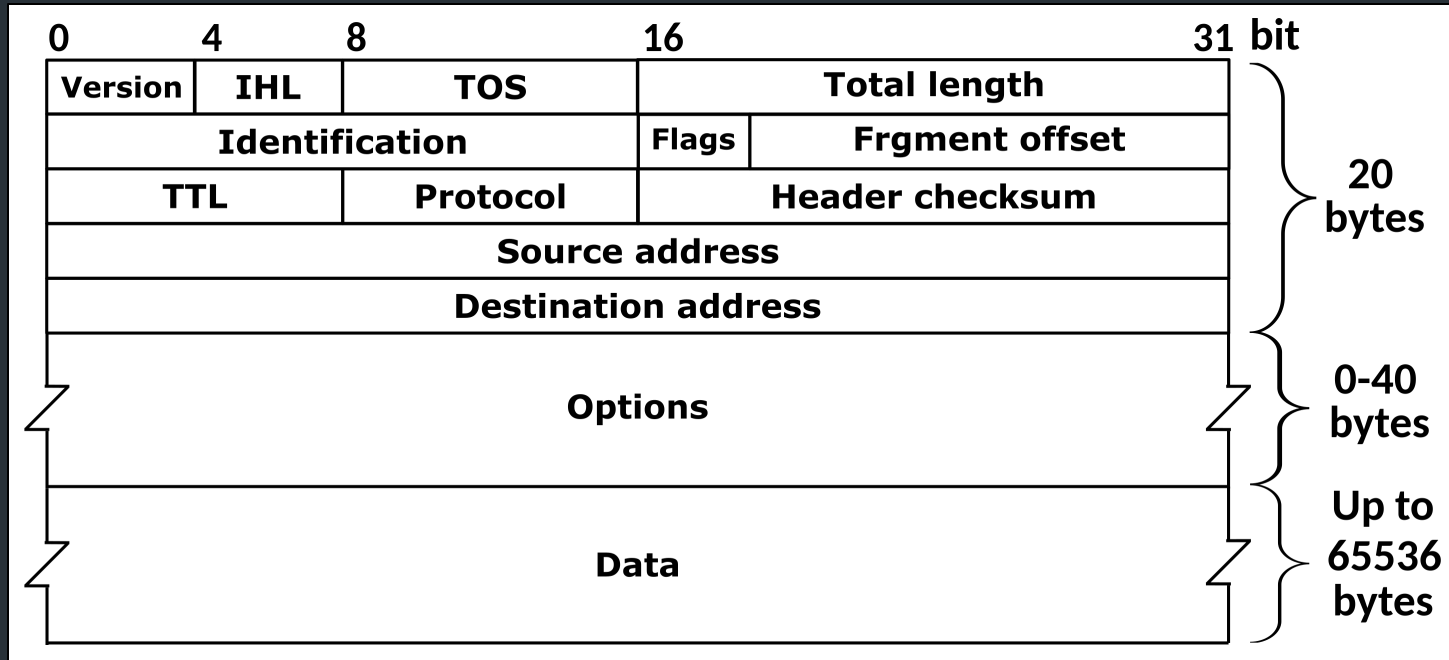
Defined by RFC 791
RFC (Request for Comment): defines network standard

Important fields

- Version: 4 for IPv4 packets, 6 for IPv6
- Destination address: used for forwarding
- TTL (time-to-live): decremented each hop
  - Can prevent forwarding loops (and do other stuff…)
- Checksum: computed over <u>header</u> (very weak!)
- Protocol identifier: describes what's in the packet
  - 6: TCP, 17: UDP, 1: ICMP, …
  - Defines the type of the payload

Less important fields

- Header length: in 32-bit units
  - \>5 implies use of IP options
  - Almost all routers ignore IP options
- Fragmentation
  - Network can fragment a packet if next link requires a small frame
  - Most routers don't fragment (or reassemble fragments)
- We won't talk about…
  - Type of Service (TOS): basic traffic classification
  - Identifier: might have special meaning on some networks

Forwarding mechanics

When an IP packet arrives at a host/router:

- Is it valid? Verify checksum over *header*

- **Is it for me?** If dest IP == your address, send to OS

- If not, where should it go?
  - Consult forwarding table => find next hop
  - Decrement TTL
  - Send packet to next hop

Traceroute

- When TTL reaches 0, router may send back an error
  - ICMP TTL exceeded
- If it does, we can identify a path used by a packet!

Traceroute example

```
[deemer@Warsprite ~]$ traceroute -q 1 google.com
traceroute to google.com (142.251.40.174), 30 hops max, 60 byte packets
 1  router1-nac.linode.com (207.99.1.13)  0.621 ms
 2  if-0-1-0-0-0.gw1.cjj1.us.linode.com (173.255.239.26)  0.499 ms
 3  72.14.222.136 (72.14.222.136)  0.949 ms
 4  72.14.222.136 (72.14.222.136)  0.919 ms
 5  108.170.248.65 (108.170.248.65)  1.842 ms
 6  lga25s81-in-f14.1e100.net (142.251.40.174)  1.812 ms
```

Traceroute example

```
[deemer@Warsprite ~]$ traceroute -q 1 amazon.co.uk
traceroute to amazon.co.uk (178.236.7.220), 30 hops max, 60 byte packets
 1  router2-nac.linode.com (207.99.1.14)  0.577 ms
 2  if-11-1-0-1-0.gw2.cjj1.us.linode.com (173.255.239.16)  0.461 ms
 3  ix-et-2-0-2-0.tcore3.njy-newark.as6453.net (66.198.70.104)  1.025 ms
 4  be3294.ccr41.jfk02.atlas.cogentco.com (154.54.47.217)  2.938 ms
 5  be2317.ccr41.lon13.atlas.cogentco.com (154.54.30.186)  69.725 ms
 6  be2350.rcr21.b023101-0.lon13.atlas.cogentco.com (130.117.51.138)  69.947 ms
 7  a100-row.demarc.cogentco.com (149.11.173.122)  71.639 ms
 8  150.222.15.28 (150.222.15.28)  78.217 ms
 9  150.222.15.21 (150.222.15.21)  84.383 ms
10  *
11  150.222.15.4 (150.222.15.4)  74.529 ms
                                    . . .
30  178.236.14.162 (178.236.14.162)  83.659 ms
```
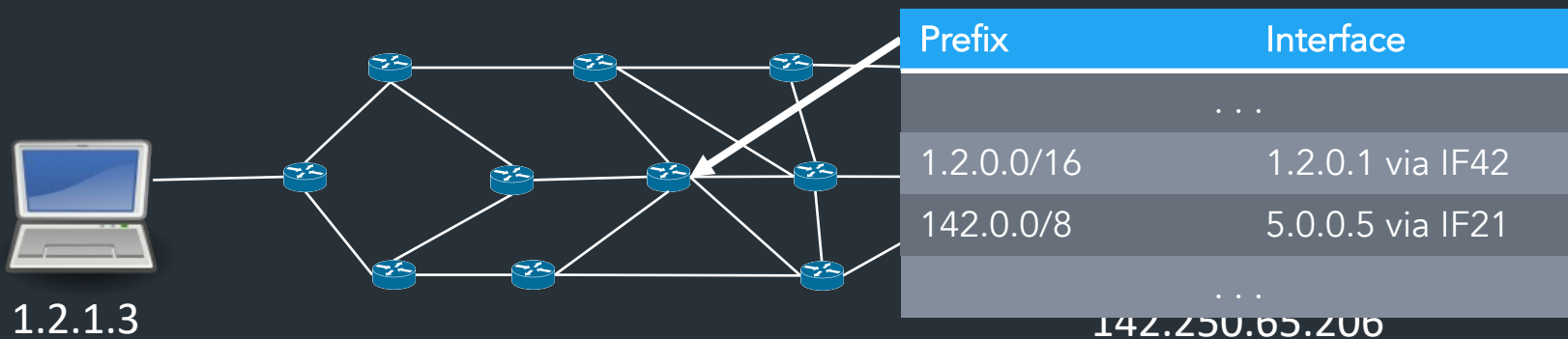
Demo: IP project

Coming up…

- ARP: Mapping IPs to MAC addresses
- How are addresses assigned?
- NAT: When it gets complicated
- Routing algorithms: how to build forwarding tables

Fill out the group preference survey for the IP project (announcement soon) by tomorrow (Sep 30) by 11:59PM

Putting it all together…

Prefix	Interface
. . .	
1.2.0.0/16	1.2.0.1 via IF42
142.0.0/8	5.0.0.5 via IF21
. . .	

1.2.1.3   142.250.65.206

- The more connected a router becomes, the more complex its forwarding table… and the more it may change!

- <u>Routing</u> algorithms: routers exchange path information to their forwarding tables (more on this later)

Goal: find the most specific (ie, longest) prefix matching the destination
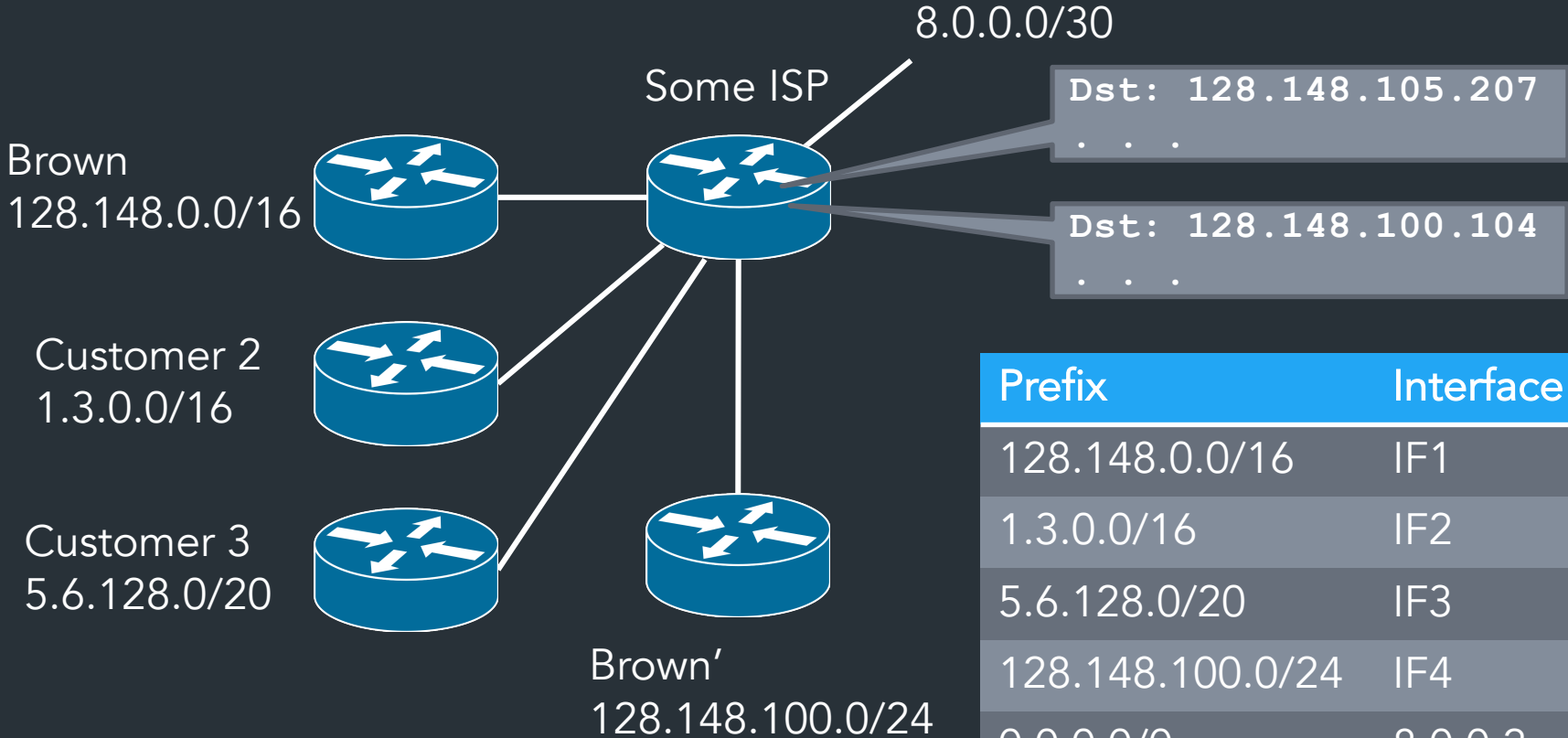
How to reach 1.2.2.100?

Prefix	Interface
1.2.1.0/24	IF1
1.2.2.0/24	IF2
0.0.0.0/0	IF0

```
1.2.2.100        00000001.00000010.00000010.01100100    Output:  IF2
                              ?=
1.2.1.0/24       00000001.00000010.00000001.xxxxxxxx
1.2.2.0/24       00000001.00000010.00000010.xxxxxxxx
0.0.0.0/0        xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
```

Longest Prefix Matching (LPM): can represent entire IP space in (small) table!

8.0.0.0/30

Some ISP

Dst: 128.148.105.207
. . .

Dst: 128.148.100.104
. . .

Brown
128.148.0.0/16

Customer 2
1.3.0.0/16

Customer 3
5.6.128.0/20

Brown'
128.148.100.0/24

Prefix	Interface
128.148.0.0/16	IF1
1.3.0.0/16	IF2
5.6.128.0/20	IF3
128.148.100.0/24	IF4
0.0.0.0/0	8.0.0.2

79

A routing table

```
R6#sh ip ro
Gateway of last resort is 108.34.215.1 to network 0.0.0.0

S*      0.0.0.0/0 [1/0] via 108.34.215.1
        10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
C          10.1.0.0/24 is directly connected, wlan-ap0
L          10.1.0.2/32 is directly connected, wlan-ap0
O IA       10.1.44.1/32 [110/1001] via 10.20.30.33, 3w4d, Tunnel0
C          10.1.48.0/24 is directly connected, Loopback0
L          10.1.48.1/32 is directly connected, Loopback0
C          10.20.30.32/31 is directly connected, Tunnel0
L          10.20.30.32/32 is directly connected, Tunnel0
        108.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          108.34.215.0/24 is directly connected, GigabitEthernet0/0
L          108.34.215.208/32 is directly connected, GigabitEthernet0/0
        172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C          172.16.98.0/24 is directly connected, Vlan98
L          172.16.98.1/32 is directly connected, Vlan98
        172.17.0.0/16 is variably subnetted, 6 subnets, 3 masks
O IA       172.17.44.0/24 [110/1001] via 10.20.30.33, 3w4d, Tunnel0
C          172.17.48.0/24 is directly connected, Vlan20
L          172.17.48.1/32 is directly connected, Vlan20
C          172.17.49.0/25 is directly connected, Vlan50
```

A routing table

```
R6#sh ip ro
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override


Gateway of last resort is 108.34.215.1 to network 0.0.0.0


S*     0.0.0.0/0 [1/0] via 108.34.215.1
       10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
C         10.1.0.0/24 is directly connected, wlan-ap0
L         10.1.0.2/32 is directly connected, wlan-ap0
O IA      10.1.44.1/32 [110/1001] via 10.20.30.33, 3w4d, Tunnel0
C         10.1.48.0/24 is directly connected, Loopback0
L         10.1.48.1/32 is directly connected, Loopback0
C         10.20.30.32/31 is directly connected, Tunnel0
L         10.20.30.32/32 is directly connected, Tunnel0
```
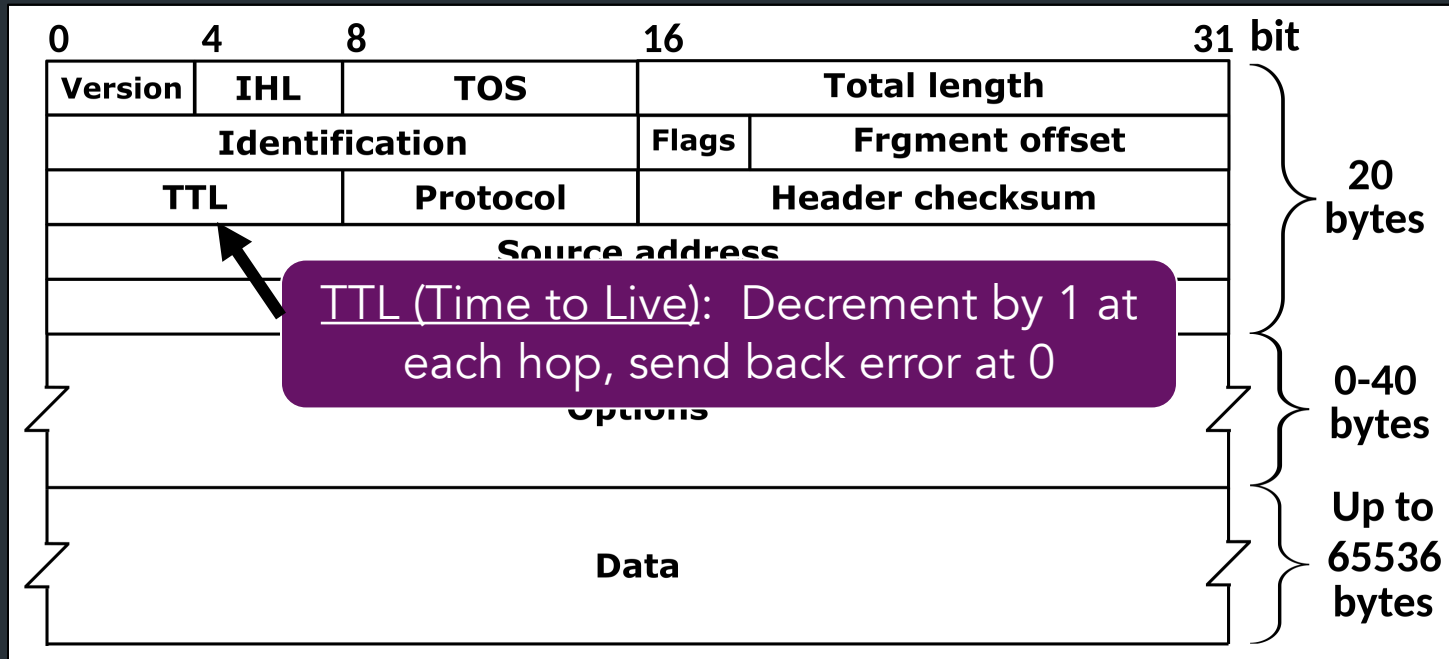
A large table

```
rviews@route-server.ip.att.net>show route table inet.0 active-path

inet.0: 866991 destinations, 13870153 routes (866991 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both


0.0.0.0/0            *[Static/5] 5w0d 19:43:09
                      > to 12.0.1.1 via em0.0
1.0.0.0/24           *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238
                        AS path: 7018 3356 13335 I, validation-state: valid
                      > to 12.0.1.1 via em0.0
1.0.4.0/22           *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238
                        AS path: 7018 3356 4826 38803 I, validation-state: valid
                      > to 12.0.1.1 via em0.0
1.0.4.0/24           *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238
                        AS path: 7018 3356 4826 38803 I, validation-state: valid
                      > to 12.0.1.1 via em0.0
1.0.5.0/24           *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238
                        AS path: 7018 3356 4826 38803 I, validation-state: valid
                      > to 12.0.1.1 via em0.0
1.0.6.0/24           *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238
                        AS path: 7018 3356 4826 38803 I, validation-state: valid
                      > to 12.0.1.1 via em0.0
```

How to avoid loops?

0	4	8	16	31 bit	
Version	IHL	TOS	Total length		20 bytes
Identification			Flags	Frgment offset	
TTL		Protocol	Header checksum		
Source address					
...					
Options					0-40 bytes
Data					Up to 65536 bytes

TTL (Time to Live): Decrement by 1 at each hop, send back error at 0

traceroute: tool to send packets with increasing TTLs
=> can learn about network paths!